BAB II

LANDASAN TEORI

2.1 Konsep Dasar Sistem

Dalam suatu perusahaan atau lembaga sistem sangatlah dibutuhkan, karena sistem dapat menunjang terhadap kinerja perusahaan atau lembaga, baik yang berskala kecil maupun besar. Supaya dapat berjalan dengan baik diperlukan kerjasama diantara unsur-unsur yang terkait dalam sistem tersebut.

Ada berbagai pendapat yang mendefinisikan pengertian sistem, seperti dibawah ini :

Menurut Jogiyanto (2005 : 1) Suatu sistem adalah suatu jaringan kerja dari prosedur-prosedur yang saling berhubungan, berkumpul bersama-sama untuk melakukan suatu kegiatan atau untuk menyelesaikan suatu sasaran yang tertentu.

Masih menurut Jogiyanto (2005 : 2) Sistem adalah kumpulan dari elemen-elemen yang berinteraksi untuk mencapai suatu tujuan tertentu.

2.1.1 Elemen Sistem

Semua sistem meliputi tiga elemen utama yaitu input, transformasi dan output. Sebagian sistem dapat mengendalikan operasi mereka sendiri yang disebut sebagai sistem lingkaran tertutup *(closed-loop system)*. Sistem lingkaran tertutup mencakup suatu mekanisme kontrol, tujuan dan lingkaran umpan balik *(feedbackloop)* disamping tiga elemen utama. Sistem yang tidak memiliki kemampuan pengendalian disebut sistem lingkaran terbuka *(open-loop system)*, dalam arti mereka berhubungan dengan lingkungan mereka. Perusahaan adalah suatu contoh sistem terbuka dan sistem lingkaran tertutup.

2.1.2 Karakteristik Sistem

Suatu sistem memiliki karakteristik atau sifat-sifat tertentu, yaitu memiliki komponen-komponen *(components)*, batas sistem *(boundary)*, lingkungan luar sistem *(environment)*, penghubung *(interprest)*, masukan *(input)*, keluaran *(output)*, pengolah *(process)* dan sasaran *(objective)* dan tujuan *(goal)*.

1. Komponen Sistem *(System Components)*

Komponen sistem atau elemen-elemen sistem dapat berupa suatu subsistem atau bagian-bagian dari sistem. Setiap sistem baik besar maupun kecil, selalu mengandung komponen-komponen atau subsistem-subsistem. Setiap subsistem mempunyai sifat-sifat dari sistem untuk menjalankan suatu fungsi tertentu dan mempengaruhi proses sistem secara keseluruhan.

2. Batas Sistem *(System Boundary)*

Batas sistem merupakan daerah-daerah yang membatasi antara satu sistem dengan sistem lainnya dengan lingkungan luarnya. Batas sistem ini memungkinkan suatu sistem dipandang sebagai satu kesatuan. Batas suatu sistem menujukkan ruang lingkup (scope) dari sistem tersebut.

3. Lingkungan Luar Sistem *(System Environment)*

Lingkungan luar sistem dari suatu sistem adalah apapun di luar batas dari sistem yang mempengaruhi operasi sistem yang dapat bersifat menguntungkan dan dapat pula merugikan sistem tersebut. Lingkungan luar yang menguntungkan merupakan energi dari sistem yang harus dijaga dan dipelihara. Sedangkan yang merugikan harus ditahan dan dikendalikan, karena akan mengganggu kelangsungan hidup sistem.

4. Penghubung Sistem

Penghubung sistem merupakan media penghubung antara satu subsistem dengan subsistem yang lainnya. Melalui penghubung ini memungkinkan sumber-sumber daya mengalir dari satu subsistem ke subsistem yang lainnya. Dengan penghubung akan terjadi interaksi antar subsistem, sehingga membentuk satu kesatuan.

5. Masukan Sistem *(System Input)*

Masukan adalah suatu energi yang dimasukkan ke dalam sistem. Masukan dapat berupa masukan perawatan (maintenance input) dan masukan sinyal (signal input). Maintenance input adalah energi yang dimasukkan supaya sistem tersebut dapat beroperasi. Contoh maintenance input di dalam sistem komputer adalah program, yang digunakan untuk mengoperasikan komputer. Sedangkan signal input adalah energi yang diproses untuk mendapatkan keluaran. Contoh signal input di dalam sistem komputer adalah data, yang dapat diolah menjadi Informasi.

6. Keluaran Sistem *(System Output)*

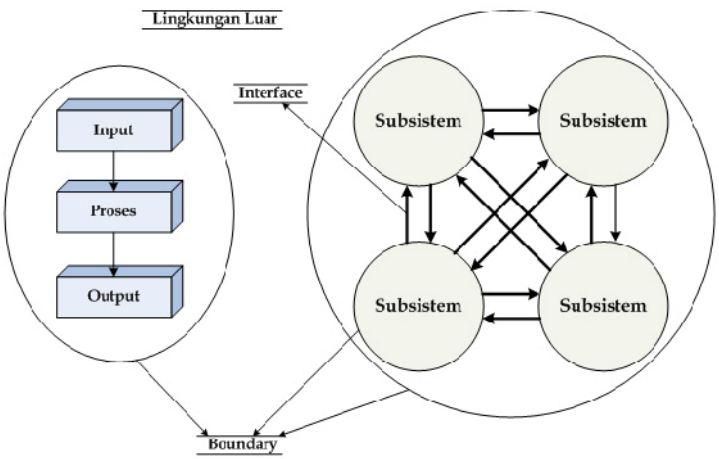
Keluaran (Output) merupakan hasil dari energi yang diolah dan diklasifikasikan menjadi keluaran yang berguna.

7. Pengolah Sistem *(System Process)*

Suatu sistem dapat mempunyai suatu bagian pengolah atau sistem itu sendiri sebagai pengolahnya, yang bertugas untuk merubah masukan menjadi keluaran.

8. Sasaran Sistem *(System Objective)*

Suatu sistem pasti memiliki tujuan (*goal*) atau sasaran (*objective*). Suatu operasi sistem akan berguna dan berhasil apabila mencapai sasaran atau tujuannya. Sasaran sistem sangat menentukan masukan yang dibutuhkan sistem dan keluaran yang akan dihasilkan sistem.



Gambar 2.1 Karakteristik Suatu Sistem

(Sumber : Jogiyanto Hartono, 2005, Analisis dan Desain, Penerbit: Andi, Yogyakarta)

2.1.3 Klasifikasi sistem

Sistem dapat diklasifikasikan dari beberapa sudut pandang, diantaranya adalah sebagai berikut :

1. Sistem diklasifikasikan sebagai sistem abstrak (*abstract system*) dan sistem pisik (*phsyikal system*).
2. Sistem diklasifikasikan sebagai sistem alamiah (*natural system*) dan sistem buatan manusia (*human made system*).

3. Sistem diklasifikasikan sebagai sistem tertentu (*deterministic system*) dan sistem tidak tentu (*probabilistic system*).
4. Sistem diklasifikasikan sebagai sistem tertutup (*closed system*) dan sistem terbuka (*open system*).

2.2 Konsep Dasar Informasi

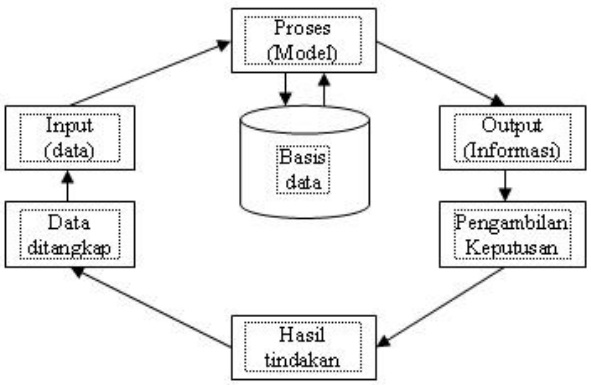
Informasi adalah data yang telah diproses dan memiliki arti atau manfaat bagi penggunanya. Informasi sebagai hasil dari pengolahan data dalam suatu organisasi sangatlah penting karena informasi merupakan landasan untuk mengambil suatu keputusan dan data merupakan sumber dari informasi.

Konsep dasar informasi menurut Jogiyanto (2005 : 8) adalah data yang diolah menjadi bentuk yang lebih berguna dan penting bagi yang menerimanya.
Dari pengertian diatas penulis dapat menyimpulkan bahwa informasi merupakan suatu hasil (*output*) dari suatu data yang diolah dengan metode pendekatan dan pengembangan tertentu.

2.2.1 Siklus Informasi

Data merupakan fakta atau kejadian yang belum berguna bagi penerimanya, sehingga perlu diolah lebih lanjut. Data diolah melalui dengan suatu metode pendekatan dan pengembangan tertentu untuk dihasilkan suatu informasi.
Data diolah sehingga menghasilkan informasi, kemudian penerima menerima informasi tersebut, membuat suatu keputusan dan melakukan tindakan, yang berarti menghasilkan suatu tindakan yang lain yang akan membuat suatu data kembali. Data tersebut akan ditangkap sebagai input, diproses kembali lewat sebuah model dan seterusnya membentuk suatu siklus. Siklus ini oleh John Burch

disebut dengan siklus informasi (*information cycle*), siklus ini disebut juga dengan siklus pengolahan data (*data processing cycle*).



Gambar 2.2 Siklus Informasi

(Sumber : Jogiyanto Hartono, 2005, Analisis dan Desain, Penerbit: Andi, Yogyakarta)

2.2.2 Kualitas informasi

Baik buruknya kualitas suatu informasi dipengaruhi oleh beberapa hal, antara lain :

1. Akurat, berarti informasi harus bebas dari kesalahan-kesalahan dan tidak bias atau menyesatkan. Informasi harus akurat karena dari sumber informasi sampai ke penerima informasi kemungkinan banyak terjadi gangguan (*noise*) yang dapat merubah atau merusak informasi tersebut.
2. Tepat pada waktunya, berarti informasi yang datang pada penerima tidak boleh terlambat. Informasi harus sesuai dengan keadaan saat itu, karena informasi yang sudah usang tidak akan mempunyai nilai lagi. Informasi merupakan landasan dalam pengambilan keputusan. Bila pengambilan keputusan terlambat, maka dapat berakibat fatal untuk organisasi. Dewasa ini mahalnya

informasi disebabkan harus cepatnya informasi tersebut didapat, sehingga diperlukan teknologi-teknologi mutakhir untuk mendapatkan, mengolah dan mengirimkannya.

3. Relevan, berarti informasi tersebut mempunyai manfaat untuk pemakainya. Relevansi informasi untuk tiap-tiap orang satu dengan lainnya berbeda. Misalnya informasi mengenai sebab-musabab kerusakan mesin produksi kepada akuntan perusahaan adalah kurang relevan dan akan lebih relevan bila ditujukan kepada ahli teknik perusahaan. Sebaliknya informasi mengenai harga pokok produksi untuk ahli teknik merupakan informasi yang kurang relevan, tetapi relevan untuk akuntan.

2.2.3 Nilai Informasi

Menurut Jogiyanto (2005 : 11), nilai dari informasi (*value of information*) ditentukan dari dua hal, yaitu manfaat dan biaya mendapatkannya. Suatu informasi dikatakan bernilai apabila manfaatnya lebih efektif dibandingkan dengan biaya mendapatkannya. Kegunaan informasi adalah untuk mengurangi hal ketidakpastian di dalam proses pengambilan keputusan tentang suatu keadaan. Akan tetapi perlu diperhatikan bahwa informasi yang digunakan didalam suatu informasi umumnya digunakan untuk beberapa kegunaan. Sehingga tidak memungkinkan dan sulit untuk menghubungkan suatu bagian informasi pada suatu masalah yang tertentu dengan biaya untuk memperolehnya. Sebagian besar informasi tidak dapat persis ditaksir keuntungannya dengan suatu nilai uang, tetapi dapat ditaksir dari nilai efektifitasnya.

2.3 Konsep Dasar Sistem Informasi

Informasi diperoleh dari sistem informasi (*information system*) atau *processing system*. Sistem informasi menurut Robert A Letch dan K. Roscoe Davis, disadur oleh Jogiyanto

> "Sistem informasi adalah suatu sistem di dalam suatu organisasi yang mempertemukan kebutuhan pengolahan transaksi harian, mendukung operasi, bersifat manajerial dan kegiatan strategi dari suatu organisasi dan menyediakan pihak luar tertentu dengan laporan-laporan yang diperlukan" (2005 : 11).

Dari pengertian diatas, sistem informasi dapat didefinisikan pula sebagai suatu kesatuan dari berbagai bagian atau elemen-elemen atau subsistem-subsistem yang saling berkaitan dan berinteraksi satu sama lainnya untuk mencapai suatu tujuan tertentu.

Sistem informasi terdiri dari komponen-komponen yang disebut dengan istilah blok bangunan (*building block*) yaitu :

1. Blok Masukan (*Input Block*)

Input mewakili data yang masuk ke dalam sistem informasi, input disini termasuk metode-metode dan media untuk menangkap data yang akan dimasukkan, yang dapat berupa dokumen-dokumen dasar.

2. Blok Model (*Model Block*)

Blok ini terdiri dari kombinasi prosedur, logika dan model matematik yang akan memanipulasi data input dan data yang tersimpan dibasis data dengan cara yang sudah ditentukan untuk menghasilkan keluaran yang diinginkan.

3. Blok Keluaran (*Output Block*)

Produk dari sistem informasi adalah keluaran yang merupakan informasi yang berkualitas dan dokumentasi yang berguna untuk semua tingkatan manajemen serta semua pemakai sistem.

4. Blok Teknologi (*Technology Block*)

Teknologi merupakan *Toolbox* dalam sistem informasi. Teknologi digunakan untuk menerima input, menjalankan model, menyimpan dan mengakses data, menghasilkan dan mengirimkan keluaran dan membantu pengendalian dari sistem secara keseluruhan. Teknologi terdiri dari 3 bagian utama, yaitu teknisi (*brainware*), perangkat lunak (*software*) dan perangkat keras (*hardware*).

5. Blok Basis Data (*Database Block*)

Basis data (*database*) merupakan kumpulan dari data yang saling berhubungan satu dengan yang lainnya, tersimpan di perangkat keras komputer dan digunakan perangkat lunak untuk memanipulasinya. Data perlu disimpan didalam basis data untuk keperluan penyedian informasi lebih lanjut.

6. Blok Kendali (*Controls Block*)

Banyak hal yang dapat merusak sistem informasi, seperti misalnya bencana alam, api, temperature, air, debu, kecurangan-kecerangan, kegagalan-kegagalan sistem itu sendiri, kesalahan-kesalahan, ketidak efisienan, sabotase dan sebagainya. Beberapa pengendalian perlu dirancang dan diterapkan untuk meyakinkan bahwa hal-hal yang dapat merusak sistem dapat dicegah ataupun bila terlanjur terjadi kesalahan-kesalahan dapat langsung cepat diatasi.

2.4 Konsep Dasar Analisis Sistem

Analisis data masukan adalah suatu analisis yang dilakukan terhadap data-data dari entitas luar yang dimasukkan kedalam sistem. Dengan tujuan untuk mendapatkan pemahaman sistem secara keseluruhan, tentang sistem yang berjalan sekarang sehingga permasalahan dapat dipecahkan dan kebutuhan pemakai sistem dapat diindentifikasi dengan benar.

Pada tahapan analisis ini menggunakan beberapa alat bantu untuk dapat menggambarkan sistem secara keseluruhan. Alat bantu yang digunakan adalah: use case, diagram sekenario dan dilanjutkan ke activity diagram.

2.4.1 *Use Case*

Use case adalah rangkaian/uraian sekelompok yang saling terkait dan membentuk sistem secara teratur yang dilakukan atau diawasi oleh sebuah aktor. *Use case* digunakan untuk membentuk tingkah-laku benda/ things dalam sebuah model serta di Realisasikan oleh sebuah collaboration. Umumnya use case digambarkan dengan sebuah elips dengan garis yang solid, biasanya mengandung nama.*Use case* menggambarkan proses system (kebutuhan system dari sudut pandang user). Secara umum use case adalah:

1. Pola perilaku system
2. Urutan transaksi yang berhubungan yang dilakukan oleh satu actor

Use case diagram terdiri dari

1. *Use case*
2. *Actors*

3. *Relationship*
4. *System boundary boxes (optional)*
5. *Packages (optional)*

2.4.2 *Sequence diagram*

Sequence diagram (diagram urutan) adalah suatu diagram yang memperlihatkan atau menampilkan interaksi-interaksi antar objek di dalam sistem yang disusun pada sebuah urutan atau rangkaian waktu. Interaksi antar objek tersebut termasuk pengguna, display, dan sebagainya berupa pesan/message.

Sequence Diagram digunakan untuk menggambarkan skenario atau rangkaian langkah-langkah yang dilakukan sebagai sebuah respon dari suatu kejadian/even untuk menghasilkan output tertentu. Sequence Diagram diawali dari apa yang me-trigger aktivitas tersebut, proses dan perubahan apa saja yang terjadi secara internal dan output apa yang dihasilkan. Diagram ini secara khusus berasosiasi dengan use case diagram. Sequence diagram juga memperlihatkan tahap demi tahap apa yang seharusnya terjadi untuk menghasilkan sesuatu didalam use case. Sequence diagram juga dapat merubah atribut atau method pada class yang telah dibentuk oleh class diagram, bahkan menciptakan sebuah class baru. Sequence diagram memodelkan aliran logika dalam sebuah system dalam cara yang visual.

Sequence diagram biasanya digunakan untuk tujuan analisa dan desain, memfokuskan pada identifikasi method didalam sebuah system. Sequence diagram biasanya dipakai untuk memodelkan :

1. Deskripsi tentang system yang ada pada sebuah / beberapa use case pada use case diagram, yang menggambarkan hubungan antara *actor* dan *use case diagram*.
2. Logika dari method (*operation, function atau procedure*).
3. Logika dari service (*high level method*).

2.4.3 *Activity Diagram*

Activity diagrams menggambarkan berbagai alir aktivitas dalam sistem yang sedang dirancang, bagaimana masing-masing alir berawal, decision yang mungkin terjadi, dan bagaimana mereka berakhir. Activity diagram juga dapat menggambarkan proses paralel yang mungkin terjadi pada beberapa eksekusi. *Activity diagram* merupakan state diagram khusus, di mana sebagian besar state adalah action dan sebagian besar transisi di-trigger oleh selesainya state sebelumnya (*internal processing*). Oleh karena itu *activity diagram* tidak menggambarkan behaviour internal sebuah sistem (dan interaksi antar subsistem) secara eksak, tetapi lebih menggambarkan proses-proses dan jalur-jalur aktivitas dari level atas secara umum. Menggambarkan proses bisnis dan urutan aktivitas dalam sebuah proses. Dipakai pada business modeling untuk memperlihatkan urutan aktifitas proses bisnis. Struktur diagram ini mirip flowchart atau Data Flow Diagram pada perancangan terstruktur. Sangat bermanfaat apabila kita membuat diagram ini terlebih dahulu dalam memodelkan sebuah proses untuk membantu memahami proses secara keseluruhan. *Activity diagram* dibuat berdasarkan sebuah atau beberapa *use case* pada *use case diagram*.

2.5 Pengertian Pengggajian

Menurut Soemarso S.R (2005 : 307) "gaji adalah imbalan kepada pegawai yang diberi tugas-tugas administratif dan pimpinan yang jumlahnya biasanya tetap secara bulanan atau tahunan".

Sedangkan menurut Mulyadi (2001 : 373) "gaji adalah pembayaran atas penyerahan jasa yang dilakukan oleh karyawan yang mempunyai jenjang jabatan manajer".

Gaji sering disebut upah, perbedaan upah dan gaji hanya terletak pada ikatan kerja dan jangka waktu penerimaan. Seseorang menerima gaji apabila ikatan kerja kuat, sedangkan seseorang menerima upah apabila ikatan kerja kurang kuat. Dilihat dari jangka waktu penerimaannya, gaji pada umumnya diberikan setiap bulan, sedangkan upah diberikan setiap hari atau setiap minggu.

Dalam bekerja seseorang akan mendapatkan penghasilan yang dapat digunakan untuk kebutuhan hidupnya. Gaji yang sering juga disebut upah merupakan sebuah penghargaan yang diberikan kepada pegawai atau karyawan sebagai balas jasa atau penghargaan atas hasil kerja seseorang.

Tunjangan adalah suatu biaya yang diberikan kepada anggota keluarga pegawai yang bersangkutan. Besarnya tunjangan sudah diatur dalam peraturan daerah yang telah ditentukan.

Gaji pokok adalah besarnya gaji yang diterima pegawai sebelum dikenakan potongan dan tunjangan – tunjangan.

2.5.1 Fungsi Penggajian

Menurut Komaruddin (2007:164), (http://books.google.co.id) fungsi gaji

bukan hanya membantu manajer personalia dalam menentukan gaji yang adil dan layak saja, tetapi masih ada fungsi-fungsi yang lain, yaitu:

1. Untuk menarik pekerja yang mempunyai kemampuan ke dalam organisasi
2. Untuk mendorong pekerja agar menunjukkan prestasi yang tinggi
3. Untuk memelihara prestasi pekerja selama periode yang panjang

2.5.2 Peranan Gaji

Menurut Poerwono (2004:124), (http://books.google.co.id) peranan gaji dapat ditinjau dari dua pihak, yaitu :

a. Aspek Pemberi Kerja (Majikan)

b. Aspek Penerima Kerja

Adapun dari pengertian diatas adalah sebagai berikut:

a. Aspek pemberi kerja (majikan) adalah manager

Gaji merupakan unsur pokok dalam menghitung biaya produksi dan komponen dalam menentukan harga pokok yang dapat menentukan kelangsungan hidup perusahaan. Apabila suatu perusahaan memberikan gaji terlalu tinggi maka, akan mengakibatkan harga pokok tinggi pula dan bila gaji yang diberikan terlalu rendah akan mengakibatkan perusahaan kesulitan mencari tenaga kerja.

b. Aspek penerima kerja

Gaji merupakan penghasilan yang diterima oleh seseorang dan digunakan untuk memenuhi kebutuhannya. Gaji bukanlah merupakan satu-satunya motivasi karyawan dalam berprestasi, tetapi gaji merupakan salah satu motivasi penting yang ikut mendorong karyawan untuk berprestasi, sehingga tinggi rendahnya gaji yang diberikan akan mempengaruhi kinerja dan kesetiaan karyawan.

2.5.3 Pengertian Sistem Penggajian

Menurut Drs. Slamet Saksono (2007:56-62) Sistem Penggajian merupakan langkah awal penerapan Manajemen Kepegawaian. Peraturan sistem penggajian harus disetujui baik oleh pimpinan maupun pegawai, sehingga tanggung jawab pelaksanaan sistem penggajian tidak banyak di tanggung oleh pemimpin saja. Penggajian diambil dari kata gaji, dimana pengertian gaji ini telah dibahas pada uraian diatas, istilah gaji biasanya digunakan untuk pegawai yang menerima hasil kerjanya dalam bentuk uang yang akan dibayarkan pada setiap minggu sekali.

Dari definisi diatas dapat disimpulkan bahwa sistem penggajian adalah seperangkat unsur yang saling berkaitan dan membentuk totalitas yang menentukan pemberian imbalan atas hasil kerja seseorang.

Sistem penggajian dibagi menjadi tiga jenis, diantaranya adalah sebagai berikut :

1. Sistem Skala Tunggal

Sistem skala tunggal adalah memberikan gaji pokok yang sama kepada pegawai yang berpangkat dengan tidak memperlihatkan sifat pekerjaan yang dilakukan dan beratnya tanggung jawab yang dipikul dalam melaksanakan pekerjaan

2. Sistem Skala Ganda

Sistem skala Ganda adalah Sistem penggajian yang menentukan besarnya gaji yang bukan saja didasarkan pada pangkat, tetapi juga didasarkan pada sifat pekerjaan yang dilakukan, prestasi kerja yang dicapai dan beratnya tanggung jawab yang dipikul dalam melaksanakan pekrjaan itu.

3. Sistem Skala Gabungan

Sistem Skala Gabungan adalah Sistem penggajian yang merupakan perpaduan antara Sistem skala Tunggal dengan sistem skala Ganda, dimana pegawai yang berpangkat dan masa kerja sama diberi gaji pokok yang sama, disamping itu diberikan tunjangan kepada pegawai yang memikul tanggung jawab yang berat.

2.5.4 Prinsip-prinsip Pemberian Gaji

Menurut Andrew F. Sikula 2002:125 Agar pegawai yang menerima gaji merasa puas, maka perlu diperhatikan prinsip-prinsip pemberian gaji sebagi berikut :

1. Gaji yang diberikan harus cukup untuk hidup pegawai dan keluarganya, dengan kata lain besarnya gaji harus memenuhi kebutuhan pokok minimum.
2. Pemberian gaji harus adil, artinya besar kecilnya tergantung pada berat ringannya kewajiban dan tanggung jawab yang dibebankan kepada pegawai yang bersangkutan.
3. Gaji harus diberikan tepat pada waktunya, gaji yang terlambat diberikan dapat menggakibatkan kemarahaan dan rasa tidak puas pegawai yang pada akhirnya akan merugikan produktifitas pegawai.
4. Besar kecilnya gaji harus mengikuti perkembangan harga pasar. Hal ini perlu diperhatikan, karena yang penting bagi para pegawai besar banyaknya uang yang diterima, tetapi berapa banyak barang atau jasa yang diperoleh dengan gaji tersebut.

5. Sistem pembayaran gaji harus mudah dipahami atau dilaksanakan, sehingga pembayaran dapat dilakukan dengan relative sangat singkat.
6. Perbedaan dalam tingkat gaji harus berdasarkan atas evaluasi jabatan yang objektif.
7. Struktur gaji harus di tinjau kembali dan mungkin harus diperbaiki apabila kondisi berubah.

2.6 Pengertian Pegawai

Pegawai menurut kamus besar Bahasa Indonesia adalah : " Orang atau seseorang yang bekerja pada suatu lembaga (perusahaan/ kantor) dan mendapatkan gaji sesuai dengan tingkatan pekerjaannya".

2.7 Pengertian Basis Data

Istilah Basis Data banyak menimbulkan interpretasi yang berbeda pada saat maraknya perangkat lunak di BASE II dan BASE II plus, sebuah berkas (dengan ekstensi DBF) biasa disebut basis data. Istilah yang tidak tepat ini meskipun telah merasuk kesejumlah pemrograman, akhirnya diluruskan kembali oleh pencipta perangkat lunak basis data yang lain.

Menurut Fatansyah (2007: 2) Basis data terdiri atas 2 kata, yaitu basis dan data. Basis kurang lebih dapat diartikan sebagai markas atau gedung tempat bersarang atau berkumpul. Sedangkan data adalah representasi fakta berita nyata yang mewakili suatu objek, suatu manusia (pegawai, siswa, pembeli, pelanggan, dll).

Basis Data sendiri dapat didefinisikan dalam sejumlah sudut pandang seperti :

1. Himpunan kelompok data (*arsip*) yang sedang berhubungan yang diorganisasi sedemikian rupa agar kelak dapat dimanfaatkan kembali dengan cepat dan mudah.
2. Kumpulan data yang saling berhubungan yang disimpan secara bersyarat sedemikian rupa dan tanpa pengulangan untuk memenuhi kebutuhan.
3. Kumpulan file / tabel / arsip yang saling berhubungan yang disimpan dalam media penyimpanan elektronik.

2.8 Perangkat Lunak Pendukung

Adapun perangkat lunak pendukung yang digunakan oleh penulis antara lain :

2.8.1 Sekilas Tentang *NetBeans IDE* 7.1.2

Menurut Sri Hartati Wijono, S.Si (2006 : 28-35) NetBeans IDE adalah IDE open source yang ditulis sepenuhnya dengan bahasa pemrograman Java menggunakan platform NetBeans. NetBeans IDE mendukung pengembangan semua tipe aplikasi Java (J2SE, web, EJB, dan aplikasi mobile). Fitur lainnya adalah sistem proyek berbasis Ant, kontrol versi, dan refactoring.Versi terbaru saat ini adalah NetBeans IDE 7.3 mengembangkan fitur-fitur Java EE yang sudah ada (termasuk Java Persistence support, EJB-3 dan JAX-WS). Sementara paket

tambahannya, NetBeans Enterprise Pack mendukung pengembangan aplikasi perusahaan Java EE 5, meliputi alat desain visual SOA, skema XML, web service dan pemodelan UML. NetBeans C / C++ Pack mendukung proyek C /C++.

Platform NetBeans memungkinkan aplikasi dibangun dari sekumpulan komponen perangkat lunak moduler yang disebut ‘modul’. Sebuah modul adalah suatu arsip Java (Java archive) yang memuat kelas-kelas Java untuk berinetraksi dengan NetBeans Open API dan file manifestasi yang mengidentifikasinya sebagai modul. Aplikasi yang dibangun dengan modul-modul dapat dikembangkan dengan menambahkan modul-modul baru. Karena modul dapat dikembangkan secara independen, aplikasi berbasis platform NetBeans dapat dengan mudah oleh pihak ketiga secara mudah dan powerfull.

2.8.2 Konsep Dasar Pemograman Berorientasi Objek (OOP)

Pemograman berorientasi objek (*object oriented programming*) merupakan pemograman yang berorientasikan kepada objek. Semua data dan fungsi di dalam paradigma ini dibungkus dalam kelas-kelas atau objek-objek. Setiap objek dapat menerima pesan, memproses data, dan mengirim pesan ke objek lainnya,

Model data berorientasi objek dikatakan dapat memberi fleksibilitas yang lebih, kemudahan mengubah program, dan digunakan luas dalam teknik piranti lunak skala besar. Lebih jauh lagi, pendukung OOP mengklaim bahwa OOP lebih mudah dipelajari bagi pemula dibanding dengan pendekatan sebelumnya, dan pendekatan OOP lebih mudah dikembangkan dan dirawat.

Istilah-istilah dalam OOP :

1. **Kelas atau *Class*** : kumpulan atas definisi data dan fungsi-fungsi dalam suatu unit untuk suatu tujuan tertentu. Sebagai contoh 'class of dog'

adalah suatu unit yang terdiri atas definisi-definisi data dan fungsi-fungsi yang menunjuk pada berbagai macam perilaku/turunan dari anjing. Sebuah class adalah dasar dari modularitas dan struktur dalam pemrograman berorientasi object. *Sebuah class s*ecara tipikal sebaiknya dapat dikenali oleh seorang non-programmer sekalipun terkait dengan domain permasalahan yang ada, dan kode yang terdapat dalam sebuah class sebaiknya (relatif) bersifat mandiri dan independen (sebagaimana kode tersebut digunakan jika tidak menggunakan OOP). Dengan modularitas, struktur dari sebuah program akan terkait dengan aspek-aspek dalam masalah yang akan diselesaikan melalui program tersebut. Cara seperti ini akan menyederhanakan pemetaan dari masalah ke sebuah program ataupun sebaliknya.

2. **Objek** : membungkus data dan fungsi bersama menjadi suatu unit dalam senuah program computer, objek adalah dasar dari modularitas dan struktur dalam sebuah program komputer berorientasi objek.
3. **Abstraksi** : kemampuan sebuah program untuk melewati aspek informasi yang diproses olehnya, yaitu kemampuan untuk memfokus pada inti. Setiap objek dalam sistem melayani sebagai model dari "pelaku" abstrak yang dapat melakukan kerja, laporan dan perubahan keadaannya, dan berkomunikasi dengan objek lainnya dalam sistem, tanpa mengungkapkan bagaimana kelebihan ini diterapkan. Proses,

fungsi atau metode dapat juga dibuat abstrak, dan beberapa teknik digunakan untuk mengembangkan sebuah pengabstrakan.

4. **Enkapsulasi** : memastikan pengguna sebuah objek tidak dapat mengganti keadaan dalam dari sebuah objek dengan cara yang tidak layak; hanya metode dalam objek tersebut yang diberi izin untuk mengakses keadaannya. Setiap objek mengakses interface yang menyebutkan bagaimana objek lainnya dapat berinteraksi dengannya. Objek lainnya tidak akan mengetahui dan tergantung kepada representasi dalam objek tersebut.
5. **Polomorfisme** melalui pengiriman pesan. Tidak bergantung kepada pemanggilan subrutin, bahasa orientasi objek dapat mengirim pesan; metode tertentu yang berhubungan dengan sebuah pengiriman pesan tergantung kepada objek tertentu di mana pesa tersebut dikirim. Contohnya, bila sebuah burung menerima pesan "gerak cepat", dia akan menggerakan sayapnya dan terbang. Bila seekor singa menerima pesan yang sama, dia akan menggerakkan kakinya dan berlari. Keduanya menjawab sebuah pesan yang sama, namun yang sesuai dengan kemampuan hewan tersebut. Ini disebut polimorfisme karena sebuah variabel tungal dalam program dapat memegang berbagai jenis objek yang berbeda selagi program berjalan, dan teks program yang sama dapat memanggil beberapa metode yang berbeda di saat yang berbeda dalam pemanggilan yang sama. Hal ini berlawanan dengan bahasa

fungsional yang mencapai polimorfisme melalui penggunaan fungsi kelas-pertama.

Dengan menggunakan OOP maka dalam melakukan pemecahan suatu masalah kita tidak melihat bagaimana cara menyelesaikan suatu masalah tersebut (terstruktur) tetapi objek-objek apa yang dapat melakukan pemecahan masalah tersebut. Salah satu pemograman pendukung OOP adalah bahasa pemograman Java.

2.8.3 Sekilas Tentang Java

Sejarah Java berawal pada tahun 1991 ketika perusahaan *Sun Microsystem* memulai *Green Project*, yakni project penelitian untuk membuat bahasa yang akan digunakan pada chip-chip embedded untuk *device intelligent customer electronic*. Bahasa tersebut haruslah bersifat *multiplatform*, tidak bergantung kepada vendor yang memanufaktur chip tersebut.

Dalam penelitiannya, projek Green berhasil membuat prototype semacam PDA (*Personal Data Assistance*) yang dapat berkomunikasi antara satu dengan yang lain dan diberi nama star 7. Ide berawal untuk membuat sistem operasi bagi Star 7 berbasis C dan C++. Setelah berjalan beberapa lama, James Gosling salah seorang team merasa kurang puas dengan beberapa karekteristik dari kedua bahasa tersebut berusaha mengembangkan bahasa lain, yang kemudian diberinama Oak, diinspirasi ketika dia melihat pohon diseberang kaca ruang kantornya. Belakangan Oak beralih nama menjadi Java.

Karena pada awalnya ditujukan untuk pemograman device kecil, Java memiliki karakteristik berukuran kecil, efisien dan portable untuk berbagai

hardware. Projek Green sempat terancam terhenti karena dalam perkembangannya, device ini belum memiliki pasar seperti yang diramalkan semula. Selanjutnya java diarahkan untuk pemograman internet. Secara kebetulan fitur-fitur java yang telah disebutkan sebelumnya sangat sesuai bagi pengembangan internet sehingga dalam beberapa tahun ke belakang ini java telah menjadi primadona untuk pemograman berbasis internet.

2.8.3.1 Keunggulan Java

Java memiliki beberapa keunggulan bila dibandingkan dengan bahasa pemograman lainnya, diantaranya:

1. **Java bersifat sederhana dan relative mudah**

   Java dimodelkan sebagian dari bahasa C++, namun dengan memperbaiki beberapa karakteristik C++, seperti mengurangi kompleksitas beberapa fitur, penambahan fungsionalitas, serta penghilangan beberapa aspek pemicu ketidakstabilan sistem pada C++. Sebagai contoh, Java mengganti konsep pewarisan lebih dari satu (*multiple inheritance)* dengan *interface,* menghilangkan konsep pointer yang sering membingungkan, otomatisasi sistem alokasi memori, dan sebagainya. Ini membuat Java menjadi relative sederhana dan mudah untuk dipelajari dibandingkan bahasa pemograman lainnya.

2. **Java berorientasi pada objek (Object Oriented)**

   Java adalah bahasa pemograman yang berorientasi objek (OOP), bukan seperti Pascal, Basic atau C yang berbasis procedural. Dalam

memcahkan masalah, java membagi program menjadi objek-objek, kemudian memodelkan sifat dan tingkah laku masing-masing. Selanjutnya Java menentukan dan mengatur interaksi antara objek satu dengan lainnya.

3. Java bersifat terdistribusi

Pada decade awal perkembangan PC (*Personal Computer*), computer hanya bersifat sebagai *workstation* tunggal, tidak terhubung satu sama lain. Saat ini sistem komputerisasi cenderung terdistribusi, mulai dari *workstation client, e-mail server, database server, web server, proxy server,* dan sebagainya.

4. **Java bersifat *Multiplatform***

Dewasa ini kita mengenal banyak platform *Operating System*, mulai dari Windows, Apple, berbagai varian UNIX dan Linux, dan sebagainya. Pada umumnya, program yang dibuat dan dikompile disuatu platform hanya bisa dijalankan di platform tersebut. Java bersifat *multiplatform*, yakni dapat diterjemahkan oleh Java Interpreter pada berbagai sistem operasi.

5. Java bersifat *MultiThread*

Thread adalah proses yang dapat dikerjakan oleh program dalam suatu waktu. Java bersifat *MultiThreaded*, artinya dapat mengerjakan beberapa proses dalam waktu yang hamper bersamaan.

2.8.4 Sekilas Tentang *MYSQL*

MYSQL adalah sebuah perangkat lunak sistem manajemen basis data (database management system) atau *DBMS* yang multitheard, multi-user, dengan 6 juta instalasi diseluruh dunia. *Mysql* ADB membuat *Mysql* menjadi perangkat lunak gratis dibawah lisensi GNU *(General Public Lincese)*. Tetapi mereka juga menjual dibawah lisensi komersial untuk kasus-kasus dimana pengunaannya tidak cocok dengan pengguanaan GPL. Tidak sama dengan proyek-proyek Apache, dimana perangkat lunak yang dikembangkan oleh komunitas umum, dan hak cipta untuk kode sumber dimiliki oleh penulisnya masing-masing, MYSQL dimiliki dan disponsori oleh sebuah perusahaan komersial Swedia MySQL AB, dimana memegang hak cipta hampir atas semua kode sumbernya. Kedua orang Swedia dan Finlandia yang mendirikan MySQL AB adalah David Axmark, Allan Larsson dan Michael ”Monty” Widenius. Selain itu MySQL memiliki beberapa keistimewaan, antara lain :

1. *Portability*

   MySQL dapat berjalan secara stabil dalam beberapa sistem operasi sepertiWindows, Linus, FreeDBS, Mac Os, X Server, Solaris, Amiga dan masih banyak lagi.

2. *Open Source*

   MySQL didistribusikan secara open source (gratis), dibawah lisensi GPL sehingga dapat digunakan secara cuma-cuma.

3. *Multiuser*

   MySQL dapat digunakan oleh beberapa user dalam waktu yang bersamaan

tanpa mengalami masalah atau konflik.

4. *Perfomance tuning*

   MySQL memiliki kecepatan yang menakjubkan dalam menangani query sederhana, dengan kata lain dapat memproses lebih banyak SQL per satuan waktu.

5. *Column type*

   MySQL memiliki tipe kolom yang sangat kompleks, seperti *signed / unsigned integer, float, double, char, text, date, timestamp*, dan lain-lain.

6. *Command* dan *Function*

   MySQL memiliki operator dan fungsi secara penuh yang mendukung perintah *Select* dan *Where* dalam *query*.

7. *Security*

   MySQL memiliki beberapa lapisan sekuritas seperti level subnetmask, nama host, dan izin akses user dengan sistem perizinan yang mendetail serta password *terenkripsi*.

8. *Scalability* dan *limits*

   MySQL mampu menangani database dalam skala besar, dengan jumlah records lebih dari 50 juta dan 60 ribu tabel serta 5 milyar baris. Selain itu batas indeks yang dapat ditampung mencapai 32 indeks pada tiap tabelnya.

9. Connectivity

   MySQL dapat melakukan koneksi dengan client menggunakan protocol TCP/IP, *Unix soket* (UNIX), atau *Named Pipes* (NT).

10. *Localisation*

    MySQL dapat mendeteksi pesan kesalahan pada client dengan menggunakan lebih dari dua puluh bahasa. Meskipun demikian, bahasa Indonesia belum termasuk didalamnya.

11. *Clients dan tools*

    MySQL memiliki *interface* (antar muka) terhadap berbagai aplikasi dan bahasa pemrograman dengan menggunakan fungsi API *(Application Programming Interface)*.

12. Struktur Tabel

    MySQL dilengkapi dengan berbagai *tool* yang dapat digunakan untuk administrasi *database*, dan pada setiap *tool* yang ada disertakan petunjuk online.

2.8.5 Xampp

XAMPP merupakan salah satu paket *installasi Apache*, PHP dan MySQL instant yang dapat kita gunakan untuk membantu proses installasi ketiga produk tersebut. Selain paket *installasi instant* XAMPP versi 1.6.3 juga memberikan fasiltias pilihan pengunaan PHP4 atau PHP5. Untuk berpindah versi PHP yang ingin digunakan juga sangat mudah dilakukan dengan mengunakan bantuan *PHP Switch* yang telah disertakan oleh XAMPP dan yang terpenting XAMPP bersifat *free* atau gratis untuk digunakan.

Sejarah singkat XAMPP, XAMPP merupakan pengembangan dari LAMP (Linux Apache, MySQL, PHP and PERL), XAMPP ini merupakan *project nonprofit* yang di kembangkan oleh *Apache Friends* yang didirikan Kai 'Oswalad'

Seidler dan Kay Vogelgesang pada tahun 2002, *project* mereka ini bertujuan mempromosikan pengunaan *Apache web server.*

Detail paket pada XAMPP Berikut detail paket *installasi* yang disertakan pada XAMPP 1.6.4 yang digunakan pada artikel ini:

a. Apache 2.2.6

b. MySQL 5.0.45

c. PHP 5.2.4 + PHP 4.4.7 + PEAR

d. PHP Switch win32 1.0

e. XAMPP *Control Version* 2.5 from www.nat32.com

f. XAMPP *Security* 1.0

g. SQLite 2.8.15

h. OpenSSL 0.9.8e

i. phpMyAdmin 2.11.1

j. ADOdb 4.95

k. Mercury Mail Transport System v4.01b

l. FileZilla FTP Server 0.9.23

m. Webalizer 2.01.10

2.8.6 *Jasper i-Report* 4.1.3

Ireport merupakan produk *OpenSource* dari *JasperReport*. *Ireport* adalah suatu *tool* yang sangat berguna untuk reporting yang paling popular di Java. *Tool* ini memudahkan kita untuk pembuatan laporan, dalam berbagai format dan memodifikasi laporan yang sangat komplek dengan *chart* format gambar, dan juga dalam bagian report lainnyaa. Didistribusikan dengan *source code* sesuai dengan *General Public License (GNU).*